

# INSTALLATION D'UN SERVEUR APACHE AVEC DOCKER

## A. Présentation de Docker

Docker est une plateforme logicielle open source qui permet de déployer et gérer facilement des applications dans des conteneurs.

Ces conteneurs sont des environnements légers et autonomes qui encapsulent une application et ses dépendances. Ces conteneurs sont isolés les uns des autres.

On peut donc très facilement installer le même serveur web avec une version de php différente dans chaque conteneur, et passer d'un conteneur à l'autre pour tester notre site web avec chaque version php.

Cela en fait une solution très prisée pour les environnements de test des développeurs.

On peut aussi l'utiliser dans le cadre d'une infrastructure à haute disponibilité comme les environnements cloud.

Plusieurs conteneurs de notre serveur seront créés, puis seront gérés par un « orchestrateur » (SWARM, KUBERNETES) pour qu'un conteneur donné prenne le relais en cas de défaillance du conteneur par défaut de notre serveur.

On peut aussi souligner que Docker est très peu gourmand en ressources matérielles par rapport à une virtualisation classique et offre plus de simplicité pour la migration des applications d'une machine physique à une autre.

Ceci est bien sûr une introduction sommaire aux possibilités d'utilisation de Docker mais cela nous permet de se faire une idée générale de ses champs d'application.

## B. Installation de Docker

### 1. Installation des dépendances :

- `sudo apt-get install apt-transport-https ca-certificates curl software-properties-common`

Cette commande installe les paquets nécessaires pour permettre l'utilisation de sources de paquets HTTPS et pour gérer les certificats SSL.

### 2. Mise à jour du système :

- `sudo apt-get update`

Cette commande met à jour la liste des paquets disponibles sur votre système.

### 3. Installation de GNUPG :

- `sudo apt-get install ca-certificates curl gnupg`

GNUPG (GNU Privacy Guard) est utilisé pour vérifier l'intégrité des paquets téléchargés.

### 4. Configuration de la clé GPG de Docker :

- `sudo install -m 0755 -d /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
sudo chmod a+r /etc/apt/keyrings/docker.gpg`

Ces commandes téléchargent et installent la **clé GPG officielle de Docker**, utilisée pour vérifier l'**authenticité** des paquets Docker.

### 5. Ajout du dépôt Docker dans la liste des sources de paquets :

- `echo "deb [arch=$(dpkg --print-architecture)] signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo "$UBUNTU_CODENAME") stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`

Cette commande ajoute le dépôt Docker à la liste des sources de paquets APT.

### 6. Mise à jour du système après l'ajout du dépôt :

- `sudo apt-get update`

**Il est important de mettre à jour la liste des paquets après avoir ajouté un nouveau dépôt.**

### 7. Installation de Docker :

- `sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`

Cette commande installe les paquets Docker, Docker CLI, Containerd, Docker Buildx Plugin et Docker Compose Plugin.

**Maintenant, Docker devrait être correctement installé sur Ubuntu.**

Pour vérifier l'installation :

`docker --version` → cette commande affiche la version installée de Docker

`grep docker /etc/group` → cette commande confirme que le groupe docker a été créée

```
haucine@Ubuntu-22:~$ docker --version
Docker version 25.0.2, build 29cf629
haucine@Ubuntu-22:~$ grep docker /etc/group
docker:x:999:
haucine@Ubuntu-22:~$
```

**Pour exécuter les commandes Docker il faut avoir les droits appropriés.**

**Ajouter l'utilisateur au groupe Docker permet de ne pas avoir à taper constamment la commande « sudo ».**

Voici les commandes à taper pour ajouter l'utilisateur au groupe docker :

`sudo usermod -aG docker haucine` → ajoute l'utilisateur haucine au groupe docker

`id haucine` → liste les groupes dont haucine fait partie

```
haucine@Ubuntu-22:~$ sudo usermod -aG docker haucine
haucine@Ubuntu-22:~$ id haucine
uid=1000(haucine) gid=1000(haucine) groupes=1000(haucine),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),122(lpadmin),135(lxd),136(sambashare) 999(docker)
```

`newgrp docker` → enregistrer les modifications

Lancer un conteneur docker « hello-world » pour tester que notre utilisateur peut le faire sans « sudo » :

```
docker run hello-world
```

```
haucine@Ubuntu-22:~$ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

haucine@Ubuntu-22:~$ █
```

**Notre installation de Docker est fonctionnelle et notre utilisateur peut lancer les commandes docker sans utiliser « sudo » à chaque fois.**

## C. Création d'un conteneur avec une image Ubuntu

Rechercher les images de Ubuntu disponibles :

```
haucine@Ubuntu-22:~$ docker search ubuntu
```

NAME	DESCRIPTION	STARS
OFFICIAL		
ubuntu [OK]	Ubuntu is a Debian-based Linux operating sys...	16844
websphere-liberty [OK]	WebSphere Liberty multi-architecture images ...	297
open-liberty [OK]	Open Liberty multi-architecture images based...	63
neurodebian [OK]	NeuroDebian provides neuroscience research s...	106
ubuntu-debootstrap [OK]	DEPRECATED; use "ubuntu" instead	52
ubuntu-upstart [OK]	DEPRECATED, as is Upstart (find other proces...	115
ubuntu/nginx	Nginx, a high-performance reverse proxy & we...	111

Télécharger l'image de Ubuntu(version officielle) :

```
haucine@Ubuntu-22:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
57c139bbda7e: Pull complete
Digest: sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
haucine@Ubuntu-22:~$
```

Lancer un conteneur avec notre image ubuntu :

```
haucine@Ubuntu-22:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
57c139bbda7e: Pull complete
Digest: sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
haucine@Ubuntu-22:~$ docker run --name SrvUbuntu -it ubuntu
root@68c1fb01a7f8:/#
```

On remarque que nous sommes dans le shell de la machine de notre conteneur :

```
haucine@Ubuntu-22:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
57c139bbda7e: Pull complete
Digest: sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
haucine@Ubuntu-22:~$ docker run --name SrvUbuntu -it ubuntu
root@68c1fb01a7f8:/#
```

**Nous venons de créer un conteneur avec une image de Ubuntu. Nous avons donc un serveur Ubuntu opérationnel.**

## D. Installation du serveur web

**Les commandes ci-dessous doivent être exécutées depuis le shell de notre conteneur.**

Elles permettent d'y installer le serveur Apache 2 ainsi que ses dépendances :

```
apt update
```

```
apt install apache2
```

```
apt install mariadb-server
```

```
apt install php libapache2-mod-php php-mysql
```

```
a2enmod php8.1
```

```
service apache2 restart
```

```
root@68c1fb01a7f8:~# service apache2 status
* apache2 is running
```

**Notre serveur web apache2 est bien fonctionnel.**